

FILEID**MTHRANDOM

L 8

MM MM TTTTTTTTTT HH HH RRRRRRRR AAAAAA NN NN DDDDDDDD 000000 MM MM
MM MM TTTTTTTTTT HH HH RRRRRRRR AAAAAA NN NN DDDDDDDD 000000 MM MM
MM MM TT HH HH RR RR AA AA NN NN DD DD DD 00 00 Mmmm Mmmm
MM MM TT HH HH RR RR AA AA NN NN DD DD DD 00 00 Mmmm Mmmm
MM MM TT HH HH RR RR AA AA NNNN NN DD DD DD 00 00 MM MM
MM MM TT HH HHHHHHHHHH RRRRRRRR AA AA NN NN DD DD DD 00 00 MM MM
MM MM TT HH HHHHHHHHHH RRRRRRRR AA AA NN NN DD DD DD 00 00 MM MM
MM MM TT HH HH RR RR AAAAAAAA NN NNNN DD DD DD 00 00 MM MM
MM MM TT HH HH RR RR AAAAAAAA NN NNNN DD DD DD 00 00 MM MM
MM MM TT HH HH RR RR AA AA NN NN DD DD DD 00 00 MM MM
MM MM TT HH HH RR RR AA AA NN NN DDDDDDDD 000000 MM MM
MM MM TT HH HH RR RR AA AA NN NN DDDDDDDD 000000 MM MM
MM MM TT HH HH RR RR AA AA NN NN

LL IIIII SSSSSSS
LL IIIII SSSSSSS
LL SS SS SS
LLLLLLLLL IIIII SSSSSSS
LLLLLLLLL IIIII SSSSSSS

MT
1-

(2)	57	DECLARATIONS
(3)	84	MTH\$RANDOM - Callable function

```
0000 1 .TITLE MTHSRANDOM - General Purpose Random Number Generator
0000 2 .IDENT /1-006/ ; File: MTHRANDOM.MAR
0000 3
0000 4
0000 5 ****
0000 6 *
0000 7 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9 * ALL RIGHTS RESERVED.
0000 10 *
0000 11 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16 * TRANSFERRED.
0000 17 *
0000 18 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20 * CORPORATION.
0000 21 *
0000 22 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24 *
0000 25 *
0000 26 ****
0000 27 *
0000 28 *
0000 29 ++
0000 30 :FACILITY: Mathematics Library
0000 31
0000 32 :ABSTRACT:
0000 33
0000 34 : This is a general random number generator. It is
0000 35 : of the multiplicative congruential type, and hence
0000 36 : is fast, although prone to certain classes of
0000 37 : non-random sequences.
0000 38
0000 39 :ENVIRONMENT: any access mode, normally user mode
0000 40 : modular, AST reentrant procedure
0000 41 --
0000 42 +
0000 43 :AUTHOR: Peter F. Conklin, CREATION DATE: 28-Dec-77
0000 44
0000 45 :MODIFIED BY:
0000 46
0000 47 : Peter F. Conklin, 1-Feb-78: VERSION 01
0000 48 : 01 - Original creation.
0000 49 : 02 - Change to LIBS rather than MTHS.
0000 50 : 03 - Change back to MTHS. Some more comments. JMT 4-Jan-78
0000 51 : 04 - Correct documentation on usage.
0000 52 : 1-005 - Put version number in standard format (three digits in
0000 53 : edit number field). JBS 16-NOV-78
0000 54 : 1-006 - Add "_" to the PSETI directive. JBS 22-DEC-78
0000 55 :-
```

0000 57 .SBTTL DECLARATIONS
0000 58
0000 59 :
0000 60 : INCLUDE FILES:
0000 61 :
0000 62 : NONE
0000 63 :
0000 64 : MACROS:
0000 65 :
0000 66 : NONE
0000 67 :
0000 68 : EQUATED SYMBOLS:
0000 69 :
0000 70 :
00000004 0000 71 SEED=4 : Only argument is the seed
0000 72
0000 73 :
0000 74 : OWN STORAGE:
0000 75 :
0000 76 : NONE
0000 77 :
0000 78 :
0000 79 :
0000 80 : PSECT DECLARATIONS:
0000 81 :
00000000 82 .PSECT _MTH\$CODE,PIC,SHR,NOWRT,LONG,EXE

```
0000 84 .SBTTL MTH$RANDOM - CALLable function
0000 85 ++
0000 86 :+ FUNCTIONAL DESCRIPTION:
0000 87 :
0000 88 : RANDOM - General Purpose Pseudo Random Number Generator
0000 89 :
0000 90 : Call the function MTH$RANDOM (seed) to obtain the
0000 91 : next pseudo-random number. The seed is updated by
0000 92 : the function automatically as a side effect. The
0000 93 : result is a floating point number that is uniformly
0000 94 : distributed in the range 0.0 inclusive to 1.0 exclusive.
0000 95 : There are no restrictions on the seed, although
0000 96 : it should be initialized to different values on
0000 97 : separate runs.
0000 98 :
0000 99 :
0000 100 :
0000 101 :
0000 102 :
0000 103 :
0000 104 :
0000 105 :
0000 106 :
0000 107 :
0000 108 :
0000 109 :
0000 110 :
0000 111 :
0000 112 :
0000 113 :
0000 114 :
0000 115 :
0000 116 :
0000 117 :
0000 118 :
0000 119 :
0000 120 :
0000 121 :
0000 122 :
0000 123 :
0000 124 :
0000 125 :
0000 126 :
0000 127 :
0000 128 :
0000 129 :
0000 130 :
0000 131 :
0000 132 :
0000 133 : CALLING SEQUENCE:
0000 134 :
0000 135 : result.wf.v = MTH$RANDOM (seed.mlu.r)
```

The algorithm used is to update the seed as:

$$\text{seed} = 69069. * \text{seed} + 1 \pmod{2^{32}}$$

and then to convert the seed to floating point.

Note, because the result is never 1.0, a simple way to get a uniform random integer selector is to multiply by the number of cases. For example if a uniform choice among 5 situations is to be made, then the following FORTRAN statement will work:

```
GO TO (1,2,3,4,5) 1+IFIX(5.*MTH$RANDOM(seed))
```

Note that the explicit IFIX is necessary before adding 1 in order to avoid a possible rounding during the normalization after the floating add.

This is a general random number generator. It is of the multiplicative congruential type, and hence is fast, although prone to certain classes of non-random sequences. This non-random behavior typically arises when considering triples of numbers generated by this method.

For more information on congruential generators, see:

Random Number Generation (pp. 1192-1197)
by G. Marsaglia

in: Encyclopedia of Computer Science
edited by Anthony Ralston
Petrocelli (New York, 1976)

0000 137 :
0000 138 : INPUT PARAMETERS:
0000 139 :
0000 140 : SEED.mlu.r Longword seed is modified each call
0000 141 :
0000 142 : IMPLICIT INPUTS:
0000 143 :
0000 144 : NONE
0000 145 :
0000 146 : OUTPUT PARAMETERS:
0000 147 :
0000 148 : NONE
0000 149 :
0000 150 : IMPLICIT OUTPUTS:
0000 151 :
0000 152 : NONE
0000 153 :
0000 154 : FUNCTION VALUE
0000 155 :
0000 156 : Returns in R0 a single-precision floating point value between
0000 157 : 0.0 inclusive and 1.0 exclusive.
0000 158 :
0000 159 : SIDE EFFECTS:
0000 160 :
0000 161 : NONE
0000 162 :
0000 163 :--

0000 0000 165 .ENTRY MTH\$RANDOM,0 ;no registers save, clear IV
0002 166
0002 167 :-
0002 168 : If this were to be placed as an inline expansion, then
0002 169 : EMUL SEED,#69069,#1,RO should replace the next two
0002 170 : instructions because this would prevent the possibility
0002 171 : of integer overflow trapping.
0002 172 :-
0002 173
04 BC 00010DCD 8F C4 0002 174 MULL2 #69069,ASEED(AP) ;update seed with multiplier
04 BC D6 000A 175 INCL ASEED(AP) ;increment seed to protect
000D 176 ; against strange seeds
000D 177
000D 178 :-
000D 179 : The next instructions convert the seed from unsigned integer
000D 180 : to floating point in the range 0.0 to 1.0 exclusive.
000D 181 :
000D 182
50 04 BC 18 08 EF 000D 183 EXTZV #8,#24,ASEED(AP),RO ;Get the most significant bits
0013 184 ; of the seed in the range
0013 185 ; 0 .. (2**24)-1
50 50 4E 0013 186 CVTLF RO,RO ;Convert to floating without
0016 187 ; rounding. The result is
0016 188 ; positive and in the range
0016 189 ; 0.0 .. (2.0**24)-1.0
0016 190
0016 191 :-
0016 192 : If this were to be placed as an inline expansion, then
0016 193 : MULF #^X00003480,RO could replace the next two instructions.
0016 194 :
0016 195
50 0C00 05 13 0016 196 BEQL 10\$;If zero, already correct
A2 0018 197 SUBW #2427,RO ;DIVF #^F2.0**24
001D 198 ; the result is now in the
001D 199 ; range 0.0 .. 1.0 exclusive
001D 200
04 001D 201 10\$: RET
001E 202
001E 203 .END

MTH\$RANDOM Symbol table

F 9
- General Purpose Random Number Generator 16-SEP-1984 01:48:11 VAX/VMS Macro V04-00
6-SEP-1984 11:26:48 [MTHRTL.SRC]MTHRANDOM.MAR;1 Page 6 (5)

MTH\$RANDOM = 00000000 RG 01
SEED = 00000004

! Psect synopsis !

PSECT NAME

Allocation	PSECT No.	Attributes											
00000000 (0.)	00 (0.)	NOPIC	USR	CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE	
0000001E (30.)	01 (1.)	PIC	USR	CON	REL	LCL	SHR	EXE	RD	NOWRT	NOVEC	LONG	

! Performance indicators !

Phase

Page faults	CPU Time	Elapsed Time
32	00:00:00.08	00:00:00.55
134	00:00:00.53	00:00:04.12
71	00:00:00.46	00:00:01.67
0	00:00:00.00	00:00:00.00
48	00:00:00.42	00:00:01.53
2	00:00:00.00	00:00:00.19
2	00:00:00.04	00:00:00.08
0	00:00:00.00	00:00:00.00
291	00:00:01.55	00:00:08.22

The working set limit was 900 pages.

1893 bytes (4 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 2 non-local and 1 local symbols.
203 source lines were read in Pass 1, producing 10 object records in Pass 2.
0 pages of virtual memory were used to define 0 macros.

Macro library statistics

Macro Library name

Macros defined

\$255\$DWA28:[SYSLIB]STARLET.MLB:2

0 GETS were required to define 0 macros.

There were no errors, warnings or information messages.

MACRO/ENABLE=SUPPRESSION/DISABLE=(GLOBAL,TRACEBACK)/LIS=LIS\$:MTHRANDOM/OBJ=OBJ\$:MTHRANDOM MSRC\$:MTHRANDOM/UPDATE=(ENHS: MTHRANDOM)

0263 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY